

### **Amendments to the Claims**

Claims 1 - 2 (canceled)

1 Claim 3 (previously presented): The method according to Claim 32, wherein the input document  
2 is a structured document.

1 Claim 4 (previously presented): The method according to Claim 3, wherein the structured  
2 document is encoded in Extensible Markup Language (“XML”).

1 Claim 5 (currently amended): The method according to Claim [[1]] 32, wherein the generated  
2 output comprises at least one object representation generated from the input document.

Claim 6 (canceled)

1 Claim 7 (previously presented): The method according to Claim 33, wherein the second syntax  
2 level is requested by specifying a schema name of a schema to which the generated output must  
3 adhere.

1 Claim 8 (previously presented): The method according to Claim 33, wherein the second syntax  
2 level is requested by specifying a schema name of a schema to be used by the validating parser  
3 when generating the output.

1 Claim 9 (previously presented): The method according to Claim 8, wherein the schema name is  
2 specified, by the application program, as a feature on an invocation of the validating parser.

Claim 10 (canceled)

1 Claim 11 (previously presented): The method according to Claim 32, wherein the first syntax  
2 level is specified in the syntax of the input document.

1 Claim 12 (previously presented): The method according to Claim 11, wherein the specification in  
2 the syntax of the input document uses a schema location construct in the input document.

1 Claim 13 (currently amended): A computer-implemented method of casting objects, comprising  
2 ~~steps of:~~

3 validating syntax elements of an input, using a validating parser, according to a first syntax  
4 level while generating output objects, from the input using the validating parser, according to a  
5 second syntax level, wherein the generating further comprises suppressing, by the validating  
6 parser, at least one of the validated syntax elements from the generated output objects in order  
7 that the generated output objects will be valid according to the second syntax level; and

8 providing the generated output objects, by the validating parser, for use by an application  
9 program.

1 Claim 14 (original): The method according to Claim 13, wherein the second syntax level is a less-

2 restrictive version of the first syntax level.

1 Claim 15 (original): The method according to Claim 13, wherein the first syntax level is a more-  
2 restrictive definition of the second syntax level.

1 Claim 16 (original): The method according to Claim 13, wherein the first syntax level is an  
2 extension of the second syntax level.

1 Claim 17 (previously presented): The method according to Claim 13, wherein the first syntax  
2 level represents an extension of the second syntax level.

1 Claim 18 (original): The method according to Claim 13, wherein the first syntax level and the  
2 second syntax level are defined using schemas.

1 Claim 19 (original): The method according to Claim 18, wherein the schema that defines the first  
2 syntax level is an extension of the schema that defines the second syntax level.

1 Claim 20 (previously presented): The method according to Claim 13, wherein the first syntax  
2 level represents a plurality of extensions to the second syntax level.

1 Claim 21 (previously presented): The method according to Claim 13, wherein the generated  
2 output objects adhere to a schema that defines the second syntax level.

1 Claim 22 (original): The method according to Claim 13, wherein the input adheres to an extended  
2 schema that defines the first syntax level.

1 Claim 23 (previously presented): The method according to Claim 22, wherein the generated  
2 output objects adhere to a base schema that is extended by the extended schema.

1 Claim 24 (previously presented): A system for applying abstraction to object markup definitions,  
2 comprising:

3 a validating parser usable by a computer;

4 first means for using the validating parser, executing on the computer, to validate syntax  
5 elements specified in an input document expressed as an object markup definition, wherein the  
6 validation is performed according to a first syntax level; and

7 second means for using the validating parser, executing on the computer, to apply  
8 abstraction to the object markup definition when generating, from the validated syntax elements,  
9 output syntax for at least one output object for use by an application program, responsive to the  
10 first means, wherein the applying of the abstraction further comprises suppressing, by the  
11 validating parser from the generated output syntax, at least one of the validated syntax elements,  
12 in order that the generated output syntax of each generated output object will be valid according  
13 to a second syntax level and wherein each of the suppressed syntax elements is valid according to  
14 the first syntax level but is not valid according to the second syntax level.

1 Claim 25 (previously presented): The system according to Claim 24, wherein the second syntax  
2 level is requested by the application program and wherein the application program then consumes  
3 at least one of the at least one generated output objects.

1 Claim 26 (currently amended): A computer program product for parsing of input, the computer  
2 program product embodied on one or more ~~computer-usable~~ computer-readable media and  
3 comprising:

4 computer-readable program code for validating, by a parser, syntax elements of an input  
5 document according to a first schema when parsing syntax of the input document; and  
6 computer-readable program code for suppressing, by the parser, at least one of the  
7 validated syntax elements when generating output from the parsed syntax of the input document,  
8 wherein each of the suppressed syntax elements is valid according to the first schema but is not  
9 valid according to a second schema for which the output is generated.

1 Claim 27 (previously presented): The computer program product according to Claim 26, wherein  
2 the first schema specifies a first syntax that is a more-restrictive version of a second syntax  
3 specified by the second schema.

1 Claim 28 (original): The computer program product according to Claim 26, wherein the first  
2 schema is defined as an extension of the second schema.

1 Claim 29 (original): The computer program product according to Claim 26, wherein the first

2 schema is defined as an extension of some intermediate schema that extends the second schema.

1 Claim 30 (currently amended): The computer program product according to Claim 26, wherein  
2 the second schema is a base schema upon which one or more extensions are based, and wherein  
3 the ~~second~~ first schema is one of the extensions and is based either directly on the base schema or  
4 on an intermediate schema that extends the base schema.

1 Claim 31 (currently amended): A computer-implemented method of providing validation and  
2 parsing for clients, comprising ~~steps of~~:  
3 providing a validating parser that enables a client to dynamically select a syntax abstraction  
4 level for use when generating output from the validating parser;  
5 obtaining an input document to be validated and parsed for the client;  
6 validating syntax elements of the input document with the provided validating parser,  
7 wherein the validation is performed according to a first syntax level to which the syntax elements  
8 of the input document are to adhere; and  
9 suppressing at least one of the validated syntax elements when generating output from the  
10 input document with the provided validating parser, for use by the client, wherein:  
11 the generated output has syntax that conforms to the syntax abstraction level that  
12 has been dynamically selected by the client;  
13 the syntax abstraction level is a less-restrictive version of the first syntax level; and  
14 each of the suppressed syntax elements is valid according to the first syntax level  
15 but is not valid according to the syntax abstraction level.

1 Claim 32 (currently amended): A computer-implemented method of applying abstraction by a  
2 validating parser, comprising ~~steps of~~:

3 using, by a validating parser, a first syntax level for validating syntax elements when  
4 parsing syntax of an input document; and

5 omitting, by the validating parser, at least one of the validated syntax elements when  
6 generating output from the parsed syntax of the input document, wherein each of the omitted  
7 syntax elements is valid according to the first syntax level but is not valid according to a second  
8 syntax level for which the output is generated.

1 Claim 33 (previously presented): The method according to Claim 32, wherein the second syntax  
2 level is requested, to the validating parser, by an application program for which the output is  
3 generated.